NASA
Technical
Paper
2760 - ReV

December 1988

Applications of the Hybrid Automated Reliability Predictor

Revised Edition

Salvatore J. Bavuso,
Joanne Bechta Dugan,
Kishor Trivedi,
Beth Rothmann,
and Mark Boyd

NASA

NASA
Technical
Paper
2760 - Rev

1988

Applications of the Hybrid Automated Reliability Predictor

Revised Edition

Salvatore J. Bavuso
Langley Research Center
Hampton, Virginia

Joanne Bechta Dugan,
Kishor Trivedi,
Beth Rothmann,
and Mark Boyd
Duke University
Durham, North Carolina

NASA
National Aeronautics
and Space Administration

Scientific and Technical
Information Division

The use of trademarks or names of manufacturers in this report is for accurate reporting and does not constitute an official endorsement, either expressed or implied, of such products or manufacturers by the National Aeronautics and Space Administration.

Abstract

The Hybrid Automated Reliability Predictor (HARP) is a software package that implements advanced reliability modeling techniques. In this paper we present an overview of some of the problems that arise in modeling highly reliable, fault tolerant systems, loosely divided into model construction and model solution problems. We then describe the HARP approach to these difficulties, which is facilitated by a technique called behavioral decomposition.

The bulk of this paper presents examples of the evaluation of some typical fault tolerant systems, including a local area network, two fault tolerant computer systems (Carnegie-Mellon University multiprocessor system C.mmp, and Software Implemented Fault Tolerance (SIFT)), and two examples of flight control systems.

Introduction

Systems with high reliability requirements are designed with a great degree of fault tolerance. These systems use extensive redundancy, both in hardware and software, have complex recovery management techniques, and are highly reconfigurable. Models used to analyze the dependability (reliability, availability, etc.) (ref. 1) of such systems will tend to be rather large so as to cover each possible configuration and condition, especially if the analyst wishes to include details of the recovery mechanisms. There are two classes of problems that arise in modeling complex, fault tolerant architectures. The first class of problems is associated with the construction of a comprehensive model of the system, and the second is associated with the solution of the model once it is formulated. We have appreciably advanced solutions in both areas and present the results of our work as they are implemented in the Hybrid Automated Reliability Predictor (HARP) under development at Duke University and sponsored by NASA Langley Research Center.

HARP is currently undergoing beta testing and is scheduled for release by NASA in 1988. (Beta testing refers to developmental software that is given to volunteer users (usually free of charge) to execute and to report problems that are discovered. Beta testing normally precedes release to the general user community.) It is written in standard FORTRAN 77 and consists of nearly 30 000 lines of code and comments and has been tested under AT&T Unix and DEC VMS. The graphics interface (written in C) runs on an IBM PC AT and produces text files that can be used to solve the system on the PC (for small systems) or that can be uploaded to a larger machine. HARP is accompanied by an introduction and guide for users. For more information about HARP, contact one of the authors.

The mathematical details of the modeling techniques used in HARP are documented elsewhere (refs. 2, 3, and 4).

Behavioral Decomposition

A common approach to modeling complex systems consists of structurally dividing the system into smaller subsystems (e.g., processors, memory units, buses), analyzing the dependability of the subsystems separately, and then combining the subsystem solutions to obtain the system solution. Dependability is a general term that includes reliability, availability, safety, etc., as special cases. A system level dependability analysis can then be effected by performing a separate analysis of each subsystem and combining the results to obtain the final solution. This structural decomposition is allowed only if the fault tolerant behaviors of the subsystems are mutually independent.

An alternative to such a structural decomposition is *behavioral decomposition* (ref. 5). We observe that the fault occurrence/repair behavior of a system is composed of relatively infrequent events, whereas fault/error-handling behavior is composed of events that occur in rapid succession once a fault has occurred. HARP allows the analyst to construct these two portions of the system model separately and combines their results together automatically. It is possible to use different model types and solution techniques for the submodels. Behavioral decomposition is a technique that facilitates both model construction and model solution.

By using behavioral decomposition, the dependability model is decomposed into a fault occurrence/repair model (FORM) and a fault/error-handling model (FEHM). The FORM contains information about the structure of the hardware redundancy, the fault arrival processes, and manual (off-line) repair. The FEHM (often called the *coverage* model) allows for the modeling of permanent, intermittent, and transient faults (ref. 6) and models the on-line recovery procedure necessary for each fault type.

We present, by way of a simple example, the model construction techniques used by HARP and discuss the HARP solution of the model. We then present models for some typical fault tolerant architectures, specifically a local area network (LAN), two fault tolerant systems, and two flight control systems.

HARP accepts the description of the system being modeled either as a fault tree or as a (continuous time) Markov chain. If the system is described as a fault tree, it is internally converted to the corresponding Markov chain. The fault/error-handling behavior is then automatically inserted into the Markov chain representation. HARP provides a variety of alternatives for the fault/error-handling model. The FORM parameters may be described in terms of a range of values or as a single-point estimate, which HARP then uses to produce a set of bounds on the time-dependent reliability measure reported. In the case of nonrepairable systems, failure distributions may be exponential or Weibull.

The Fault Occurrence/Repair Model (FORM)

Suppose we wish to model a computer consisting of three processors and two shared memories communicating over a shared bus. The system is operational as long as one processor can communicate with one of the memories. We can describe the system structure model to HARP in one of two ways: as a fault tree or as a Markov chain. A fault tree representation of a system is often more concise than the corresponding Markov chain, but a Markov chain can model more complex behavior. Sequence dependent failures and repairable systems (where there is not an independent repair crew for each component) can be modeled with a Markov chain but not a fault tree. Both ways will be described for this example.

Fault tree representation of the FORM. A fault tree is a model that graphically and logically portrays the various combinations of events occurring in a system that may lead to system failure (ref. 7). The fault tree is structured so that the the top event, system failure, is expressed as a logical function of the events that cause it. The fundamental logic gates of fault trees allowed by HARP are the AND gate, the OR gate, and the K/N gate. The input events to each logic gate may also be outputs of other logic gates at a lower level; each event is decomposed into lower events until the basic causes of faults are reached. These basic events appear as circles on the bottom of the fault tree.

If a failure represented by the logic gate is caused by the occurrence of one or more events, these events are input to an OR gate. If the failure represented by the gate occurs only when all of a set of events occur, these events are input to an AND gate instead. A K/N gate is used when the occurrence of K or more of the N possible events cause failure.

Figure 1 is the fault tree representation of the 3-processor, 2-memory, 1-bus system. The abbreviation for the combined basic event $i * j$ represents i replications of component type j. A model can be described to HARP graphically by maneuvering a cursor on the screen and placing the appropriate nodes and arcs. If a graphics device is not available, the fault tree can be described textually by means of an interactive menu-driven interface. The textual description file is simple and can be edited for minor changes. The textual fault tree description file for this example is shown in table 1.

Markov chain representation of the FORM. A Markov chain is a state-transition diagram in which each state depicts a particular operational configuration of the system. Transitions between states signify components failing and/or being repaired. Additionally, failure states (Fj, $j = 1, 2, 3, \ldots,$ total number of component types) represent various configurations that fall below the minimum necessary for an operational system. Figure 2 shows the Markov chain representation of the system whose fault tree is shown in figure 1. The states are labeled with an ordered triple, where the first element of the triple denotes the number of operational processors, the second element of the triple denotes the number of operational memories, and the third element denotes the state of the bus. The Markov chain can be entered either graphically or textually. The text file that describes the Markov chain in figure 2 is shown in table 2.

An arc is labeled with an expression containing transition rates and constant (integer or real) multipliers. Failure rate transitions are denoted by a single failure "rate" variable (i.e., λ or μ) even though HARP does not require the failure rates to be constant. The specification of the failure distribution as either exponential or Weibull is done at run time. Rates may be connected by the operations of addition, subtraction, and multiplication. In our 3-processor, 2-memory, 1-bus example, the labels are of the form: constant × failure rate. An arc between states (i,j,k) and $(i-1,j,k)$ is labeled with the value $i\lambda$ (where λ is the failure rate of processors). Likewise, an arc between states (i,j,k) and $(i,j-1,k)$ is labeled with the value $j\mu$ (where μ is the failure rate of memories). Although most transitions are of this type, transitions between arbitrary pairs of states with more general labels are permitted.

If the modeler desires a comparison of the probabilities of exhaustion of n different component types, then the failure state can be divided into n different failure states. These "exhaustion of redundancy" failure states are labeled Fj, where j represents the index of the component type. In figure 2, state F1 represents exhaustion of the processor cluster, state F2 represents exhaustion of the memories, and state F3 represents failure of the bus.

Conversion of the fault tree to a Markov chain. If a fault tree is input to HARP, it is internally converted to a Markov chain for solution. All possible occurrences of basic events that leave the system operational are enumerated; each combination becomes a state in the Markov chain (ref. 8). The fault tree that is shown in figure 1 is converted automatically to the Markov chain that is shown in figure 2. The internal representation of the FORM is the same, regardless of whether it was first described as a Markov chain or fault tree. Once the Markov chain representation of the system is determined, HARP can automatically incorporate the fault/error-handling model, since the Markov representation can model such dynamic behavior.

The advantage of allowing a fault tree description of the system is that the modeler need not perform the tedious task of determining the Markov chain representation of a system that can be described as a fault tree. Very often, a relatively simple fault tree can give rise to a very large state space in the corresponding Markov chain. The modeler can use the parsimony of the fault tree representation of the system to generate the state space of the Markov chain automatically and then make adjustments to the Markov chain as needed. The advantage of allowing the Markov chain representation of a system is that a Markov chain can model much more complex system behavior than a fault tree. For example, a Markov chain can model repairable systems and sequence dependent failures.

The Fault/Error-Handling Model (FEHM)

We now concentrate on detailed modeling of the behavior of the system when a fault occurs. HARP allows the analyst to parameterize any of a number of fault/error-handling models, the general structure of which is shown in figure 3. The entry point to the model signifies the occurrence of a fault, and the three exits signify three possible outcomes. The transient restoration exit R represents the correct recognition of and recovery from a transient fault/error. A transient is usually caused by external or environmental factors, such as excessive heat or a glitch in the power line. It is generally believed that most faults and errors are transient. Successful recovery from a transient fault restores the system to a consistent state without discarding any components, for example by retrying an instruction or rolling back to a previous checkpoint. Reaching this exit successfully requires timely detection of an error produced by the fault, performance of an effective recovery procedure, and the swift disappearance of the fault (the cause of the error).

The permanent coverage exit C denotes the determination of the permanent nature of the fault and the successful isolation and removal of the faulty component. The single-point failure exit S is reached when a single fault causes the system to crash. This generally occurs when an undetected error propagates through the system, or when the faulty unit cannot be isolated and thus the system cannot be reconfigured. We will describe each of the FEHM's in the applications section.

The HARP FEHM's offer the user a wide selection of models. Some are more useful during system design (e.g., the automated reliability interactive estimation system (ARIES) transient fault model; the computer-aided reliability estimation, third generation, (CARE III) single-fault model; and the extended stochastic Petri net (ESPN) model), whereas others would be useful after system fabrication (e.g., probabilities and distributions, and probabilities and moments). During the system design phase, many of the FEHM parameters are unknown. Usually, best engineering "guestimates" are used to make parametric sensitivity studies. If it is shown that certain parameters have little effect on the system dependability, these parameters are dropped from further consideration, although others may be flagged for more accurate measurement. The use of logic fault simulators has been successfully employed to provide fault and error latency measurements for systems as complex as avionic multicomputers (ref. 9). The data collected from such fault latency experiments can be easily integrated into a FEHM by using the "probabilities and empirical data" model. After hardware is fabricated, direct measurement becomes possible, which makes the use of the "probabilities and moments" or "probabilities and distributions" FEHM's attractive. In the overall system model, different types of FEHM's can be mixed.

As an example of a FEHM for the memory subsystem of figure 1, assume that single-bit memory errors (which are 98 percent of all memory faults) can be masked and that faults that affect more than one memory bit are 95 percent detectable. Upon detection of a multiple memory error, the affected portion of memory is discarded, the memory mapping function is updated, and the necessary information is reloaded from a previous checkpoint and updated to represent the current state of the system. The first two moments of the time to perform this recovery have been determined by experiment to be 0.45 and 0.25 (time scale in seconds). (If, in successive experiments, recovery times are $T_1, T_2, \ldots, T_k$, then the mean is given by $\frac{1}{k}\sum_{j=1}^{k} T_j$ and the second moment is given by $\frac{1}{k}\sum_{j=1}^{k} (T_j)^2$.

using a six-tuple $(n_1, n_2, n_3, n_4, n_5, n_6)$, where each n_i $(i = 1, 2, \ldots, 6)$ represents the number of operational nodes on each AWC$_i$. The Markov chain representing this system is shown in figure 9.

When a node fails on AWC$_j$, a transition occurs to the state where each n_i $(i \neq j)$ is unchanged and $n_j = n_j - 1$. Node repair transitions reverse this process unless $n_j = 0$ (see below).

When AWC$_j$ itself fails, a transition is made to the state where each n_i $(i \neq j)$ is unchanged and $n_j = 0$. Thus, when $n_j = 0$, either the AWC has failed or four node failures have occurred. The repair transition from a state where $n_j = 0$ leads to the corresponding state where $n_j = 4$ and all n_i $(i \neq j)$ are unchanged.

The state space is greatly reduced by the following two considerations. First, because more than seven failures represent system failure, $n_1 + n_2 + \ldots + n_6 > 16$ for all nonfailure states. Thus, all states not meeting this criterion are lumped into the state called FAIL. Second, the n_i's are not ordered. Thus, states such as (444443), (444434), (444344), (443444), etc., can be aggregated into a single state.

Taking advantage of these simplifications, the Markov description of this 24-node token-ring network requires 38 states and 188 transitions. Allowing the network to fail only when all nodes are down increases the HARP model to 210 states with 1302 transitions. The failure probabilities and the predicted availability for the 24-node network are listed in table 7.

Modeling Several Fault Tolerant Systems

Analysis of Carnegie-Mellon University Multi-miniprocessor

The Carnegie-Mellon University multiprocessor system C.mmp consists of up to 16 DEC PDP-11 processors that can communicate with up to 16 shared memory ports by using a crossbar switch (ref. 17). By a parts count method, the failure rate of an individual processor was found to be 68.9 per million hours, that of a memory port to be 224 per million hours, and that of the crossbar switch to be 202 per million hours (ref. 18). The system is operational whenever K or more processors, K or more memories, and the switch are operational.

This example was run on HARP (assuming perfect coverage), and the system unreliability was mainly caused by the crossbar switch for small K, but by the memory subsystem for large K. Figure 10 is a plot of the system reliability as a function of mission time for K from 4 to 16 by increments of 2. Fixing $K = 4$, we then ran the same problem with Weibull failure rates with the shape parameter $\alpha = 0.5$ and the scale parameter λ_i adjusted so that the MTTF for each component was preserved under the exponential and Weibull cases. We also attached a constant coverage factor of 0.9 to processor failures, whereas failures in other components were assumed to be perfectly covered. The results are shown in figure 11. The effect on the system unreliability of the time-dependent failure rates is much more pronounced than that of imperfect coverage.

Analysis of Software Implemented Fault Tolerance (SIFT)

The SIFT system was designed by the Stanford Research Institute and built by Bendix. The system is a prototype of a flight control computer for fly-by-wire aircraft with ultrahigh reliability requirements (ref. 19). We present a simple reliability model of the SIFT system (with n processors) in figure 12(a), where state (h, d, f) with $h \leq d \leq f$ represents the configuration with f faults in individual processors, d of which have been detected, and h of these have been handled by reconfiguration. A state with $d \neq f$ represents undetected latent faults, and a state with $h \neq d$ represents a situation in which the system is reconfiguring. Following the method of reference 20, we assume the detection to be perfect and instantaneous, whereas the handling process takes a constant time τ. Since the detection rate δ is infinite, states such as (0,0,1) are instantaneous and can be removed. The resulting FORM for a 5-processor SIFT system is shown in figure 12(b). In the framework of HARP, we model the handling states by the "probabilities and distributions" FEHM: the associated probability and distribution to the C exit being 1 and constant τ, respectively. By using $\lambda = 10^{-4}$ to be the processor failure rate and for $\tau = 100$ ms and 1 s, we obtain the results shown in table 8.

Modeling Ultrareliable Flight Control Systems

Advanced Reconfigurable Computer System (ARCS)

The ARCS design (ref. 21) depicted in figure 13 is of fundamental importance, as it provided the basis for the design of the digital McDonnell Douglas primary flight control system (PFCS) onboard the first production F/A-18 high-performance fighter aircraft. The original ARCS design was based on the assumption that high coverage was obtainable, and on that condition, reconfiguration to the simplex mode was adequate to provide sufficient reliability for short-term application aboard commercial aircraft (the designed application of ARCS). Because it was later determined that adequate coverage to allow reconfiguration to the simplex mode for the F/A-18 PFCS was not within the state of the art, the PFCS did not incorporate simplex reconfiguration (ref. 22). This example problem takes another look at the original ARCS short-term configuration, which permitted reconfiguration to the simplex mode, using the HARP capability. The HARP extended stochastic Petri net (ESPN) fault/error-handling model is more sophisticated and more realistic than the model used in the ARCS assessment. The HARP ESPN model (refs. 3, 4, 23, and 24), shown in figure 14, captures three aspects of fault recovery: physical fault behavior, transient recovery, and permanent recovery. The fault behavior model captures the physical status of the fault, such as whether the fault is active or benign (if intermittent), active and either producing or not producing errors (if permanent), or whether the fault still exists (if transient). Once the fault is detected, it is temporarily assumed to be transient, and an appropriate recovery procedure may commence. The transient recovery procedure may be attempted more than once. If the detection and recovery cycle is repeated too many times, a permanent recovery procedure (isolation and reconfiguration) is invoked. If the permanent recovery is successful, the system will be again operating correctly, although in an operationally degraded state.

The user inputs to this model are the distribution of time for each activity and the probabilities of error detection, fault detection, fault isolation, and reconfiguration. (The distributions need not be exponential.) The user must specify the number of attempts at transient recovery, the percentage of faults that are transient, and since this model is simulated for solution, the confidence level and percent error allowed.

The following analysis uses the same ARCS fault occurrence model (fig. 15) as that used in reference 21. The basic events labeled with an s represent the sensors, c the computers, h the hydraulic systems, and v the servos. The fault tree shows that the hydraulic system is cross-strapped to the servos, so that all three must fail (AND gate) to cause total loss of the servo capability. The device failure rates (all have exponentially distributed times to failure) and detection probabilities (coverages) are listed in reference 21 and summarized in table 9.

In the original reliability analysis, perfect coverage was assumed while in the triple modular redundant (TMR) configuration. Coverage failures could only occur if there were two or fewer units remaining in a stage. To duplicate this analysis, we wish to override the inclusion of a coverage factor for first failures. There are two ways in which this can be done. First, we could construct the full Markov model of the system, including the desired coverage factors where desired, and declare that the model is to be solved "as is." In this way, any arbitrary Markov chain can be solved. However, the specification of 625 states and over 2100 transitions, even graphically, is not a pleasant task. An easier method is to have HARP generate the Markov chain that corresponds to the fault tree, and for each transition emanating from a state with three operable units (of the same type) we simply specify that no FEHM is desired for that transition. If a transition is labeled with rate "$3*\lambda$;", we change it to "$3*\lambda$:NONE;". (Or we can change it to "$3*\lambda$:FEHM.1" to specify that the FEHM file described in the file named "FEHM.1" should apply to the transition, rather than what is listed in the dictionary.) The solution of the resulting model agrees with the original ARCS reliability assessment (ref. 21).

The ARCS designers were concerned that the coverage was not sufficient to allow reconfiguration to the simplex mode and required the system to run in the TMR mode. Indeed, the single-point failure probability is the greatest contributor to system unreliability. Since fault and error detection is perfect for a TMR configuration (assuming independence), they assumed that the reliability would be improved. For the sake of comparison, we solved the TMR model. This simply required changing the AND gates in figure 15 to "2/3" gates. There was no improvement in the reliability, since the probability of exhaustion of components increases dramatically if two units are required. We then decided to replace the constant coverages in the original model with the ESPN model, using the constant coverage values for the detection probabilities, and to assume that all near-coincident faults are fatal. The text file containing the parameters used for the FEHM is shown in table 10.

Table 11 depicts the results of the HARP run for the original design (allowing reconfiguration to simplex mode), the TMR configuration, and the original design which includes the ESPN model. We list, for a 10-hour mission, the failure probabilities attributable to exhaustion of redundancy, single-point failure, and near-coincident faults. The difference in unreliability between the model with constant coverage and the ESPN model can be attributed to the consideration of transient restoration in the ESPN model rather than assuming that all faults are permanent and result in reconfiguration.

Fault Tolerant Jet Engine Control System

This example demonstrates the capability of HARP to solve very large models. The system consists of 20 components distributed among 7 stages. (A stage is a set of redundant components with the same failure distribution.) The fault tree for this jet engine control system is shown in figure 16, and the stages are detailed in table 12. Since the redundant components within a stage function differently, each single element of a stage becomes a basic event in the fault tree, giving a total of 20 basic events. This somewhat simple looking fault tree is then converted to a Markov chain that contains 24 533 states and over 335 000 transitions between states. The Markov chain would be at least an order of magnitude larger if the full chain (including the coverage submodel explicitly) were generated.

The FEHM used for this example was the Markov version of the CARE III single-fault model (ref. 25), shown in figure 17. The CARE III coverage model, like the HARP model, can be used to model permanent, transient, and intermittent faults. In the active state, a fault is both detectable and capable of producing an error. Once an error is produced, if it is not detected, it propagates to the output and causes system failure. If the fault (error) is detected, the faulty element is removed from service with probability P_A or P_B. With the complementary probabilities, the element is returned to service after the detection of the fault. This action is based on the belief that the detected fault was transient. Note that both states A_D and B_D are "instantaneous" (i.e., zero holding time) states. In table 13, we compare the unreliability of this model assuming perfect and imperfect coverage.

Concluding Remarks

We have presented an overview of the modeling techniques used in HARP (Hybrid Automated Reliability Predictor), which is under development at Duke University and sponsored by Langley Research Center. The HARP approach to dependability analysis is characterized by a *behavioral decomposition* of the model, which facilitates both the model construction and model solution. In this paper, we have demonstrated how HARP can be used to model a variety of fault tolerant systems, models of which range from small and simple to very large (25 000 states even after the reduction afforded by behavioral decomposition).

Acknowledgments

We wish to thank Robert Geist and Mark Smotherman of Clemson University for their work in the original design of the HARP. From Duke, we thank Earl Smith for the LAN example, Malathi Veeraraghavan for her help with the C.mmp example, and Herve Tardif for his work on the interface and on the engine. We also would like to thank Anna Martensen of Kentron (at Langley Research Center) and the Beta test site users for their feedback on earlier versions of HARP.

NASA Langley Research Center
Hampton, VA 23665-5225
September 24, 1987

References

1. Laprie, Jean-Claude: Dependable Computing and Fault Tolerance: Concepts and Terminology. *The Fifteenth Annual International Symposium on Fault-Tolerance Computing—Digest of Papers,* IEEE Catalog No. 85CH2143-6, IEEE Computer Soc., 1985, pp. 1–11.
2. Dugan, Joanne Bechta; Trivedi, Kishor S.; Smotherman, Mark K.; and Geist, Robert M.: The Hybrid Automated Reliability Predictor. *AIAA J. Guid., Control and Dynamics,* vol. 9, no. 3, May–June 1986, pp. 319–331.
3. Geist, Robert; Trivedi, Kishor; Dugan, Joanne Bechta; and Smotherman, Mark: Design of the Hybrid Automated Reliability Predictor. *Proceedings IEEE/AIAA 5th Digital Avionics Systems Conference,* IEEE Catalog No. 83CH1839-0, Inst. of Electrical and Electronics Engineers, Inc., c.1983, pp. 16.5.1–16.5.8.
4. Trivedi, Kishor; and Dugan, Joanne Bechta: Hybrid Reliability Modeling of Fault-Tolerant Computer Systems. *Comput. & Electr. Eng.,* vol. 11, no. 2/3, 1984, pp. 87–108.
5. Trivedi, Kishor S.; and Geist, Robert M.: Decomposition in Reliability Analysis of Fault-Tolerant Systems. *IEEE Trans. Reliab.,* vol. R32, no. 5, Dec. 1983, pp. 463–468.
6. Siewiorek, Daniel P.; and Swarz, Robert S.: *The Theory and Practice of Reliable System Design.* Digital Press, c.1982.
7. Barlow, R. E.; and Lambert, H. E.: Introduction to Fault Tree Analysis. *Reliability and Fault Tree Analysis—Theoretical and Applied Aspects of System Reliability and Safety Assessment,* Richard E. Barlow, Jerry B. Fussell, and Nozer D. Singpurwalla, eds., Soc. of Industrial and Applied Mathematics, 1975, pp. 7–35.
8. Boyd, Mark A.: *Converting Fault Trees to Markov Chains for Reliability Prediction.* M.S. Thesis, Duke Univ., 1986.
9. McGough, John G.; Swern, Fred L.; and Bavuso, Salvatore: New Results in Fault Latency Modelling. *AIAA Guidance and Control Conference—A Collection of Technical Papers,* c.1983, pp. 882–885. (Available as AIAA-83-2303.)
10. Makam, Srinivas V.; and Avižienis, Algirdas: ARIES 81: A Reliability and Life-Cycle Evaluation Tool for Fault-Tolerant Systems. *FTCS 12th Annual International Symposium, Fault-Tolerant Computing—Digest of Papers,* IEEE Catalog No. 82CH1760-8, IEEE Computer Soc., 1982, pp. 267–274.
11. Ng, Ying-Wah; and Avižienis, Algirdas: A Model for Transient and Permanent Fault Recovery in Closed

Fault-Tolerant Systems. *Proceedings—FTCS-6*, IEEE Catalog No. 76CH1094-2 C, Inst. of Electrical and Electronics Engineers, Inc., c.1976, pp. 182–188.

12. McGough, John; Smotherman, Mark; and Trivedi, Kishor: The Conservativeness of Reliability Estimates Based on Instantaneous Coverage. *IEEE Trans. Comput.*, vol. C-34, no. 7, July 1985, pp. 602–609.
13. Shampine, L. F.; and Watts, H. A.: Global Error Estimation for Ordinary Differential Equations. *ACM Trans. Math. Softw.*, vol. 2, no. 2, June 1976, pp. 172–186.
14. Smotherman, Mark; Geist, Robert M.; and Trivedi, Kishor S.: Provably Conservative Approximations to Complex Reliability Models. *IEEE Trans. Comput.*, vol. C-35, no. 4, Apr. 1986, pp. 333–338.
15. Dixon, R. C.; Strole, N. C.; and Markov, J. D.: A Token-Ring Network for Local Data Communications. *IBM Syst. J.*, vol. 22, no. 1/2, 1983, pp. 47–62.
16. Andrews, Don W.; and Schultz, Gary D.: A Token-Ring Architecture for Local-Area Networks: An Update. *Proceedings Computer Networks—COMPCON 82 Fall*, IEEE Catalog No. 82CH1796-2, IEEE Computer Soc., c.1982, pp. 615–624.
17. Wulf, William A.; and Bell, C. G.: C.mmp—A Multi-Mini-Processor. *AFIPS Conference Proceedings, Volume 41, Part II, 1972 Fall Joint Computer Conference*, AFIPS Press, c.1972, pp. 765–777.
18. Siewiorek, D. P.: Multiprocessors: Reliability Modelling and Graceful Degradation. *System Reliability and Integrity, State of the Art Report*, Infotech International (Maidenhead, England), 1978, pp. 281–307.
19. Wensley, John H.; Lamport, Leslie; Goldberg, Jack; Green, Milton W.; Levitt, Karl N.; Melliar-Smith, P. M.; Shostak, Robert E.; and Weinstock, Charles B.: SIFT: Design and Analysis of a Fault-Tolerant Computer for Aircraft Control. *Proc. IEEE*, vol. 66, no. 10, Oct. 1978, pp. 1240–1255.
20. Goldberg, Jack; Kautz, William H.; Melliar-Smith, P. Michael; Green, Milton W.; Levitt, Karl N.; Schwartz, Richard L.; and Weinstock, Charles B.: *Development and Analysis of the Software Implemented Fault-Tolerance (SIFT) Computer.* NASA CR-172146, 1984.
21. Bjurman, B. E.; Jenkins, G. M.; Masreliez, C. J.; McClellan, K. L.; and Templeman, J. E.: *Airborne Advanced Reconfigurable Computer System (ARCS).* NASA CR-145024, 1976.
22. Harschburger, H. E.; Glaser, B.; and Hammel, J. R.: Backup Modes for the F/A-18 Digital Flight Control System. *Proceedings Sixth Digital Avionics Systems Conference*, American Inst. of Aeronautics and Astronautics, 1984, pp. 108–115. (Available as AIAA-84-2622.)
23. Dugan, Joanne Bechta; Trivedi, Kishor S.; Geist, Robert M.; and Nicola, Victor F.: Extended Stochastic Petri Nets: Applications and Analysis. *Performance '84—Models of Computer System Performance*, E. Gelenbe, ed., Elsevier Science Publ. Co., Inc., c.1984, pp. 507–519.
24. Trivedi, Kishor; Dugan, Joanne Bechta; Geist, Robert; and Smotherman, Mark: Modeling Imperfect Coverage in Fault-Tolerant Systems. *Proceedings of the Fourteenth International Conference on Fault-Tolerant Computing,* Inst. of Electrical and Electronics Engineers, Inc., c.1984, pp. 77–82.
25. Stiffler, J. J.; and Bryant, L. A.: *CARE III Phase II Report—Mathematical Description.* NASA CR-3566, 1982.

Table 1. HARP-Generated Textual File Representing the Fault Tree of Figure 1

```
NODE  1:  TYPE BASIC,  3  OF COMPONENT  1
NODE  2:  TYPE BASIC,  2  OF COMPONENT  2
NODE  3:  TYPE BASIC,  1  OF COMPONENT  3
NODE  4:  TYPE  AND       ,  1   INPUTS:  1
NODE  5:  TYPE  AND       ,  1   INPUTS:  2
NODE  6:  TYPE  OR     ,   3   INPUTS:  3  4  5
NODE  7:  TYPE FBOX, INPUT:   6
```

Table 2. HARP-Generated Textual Description of Markov Chain of Figure 2

```
 TEXTUAL
3,2,1   2,2,1   3*LAMBDA;          3,1,1   2,1,1   3*LAMBDA;
2,2,1   1,2,1   2*LAMBDA;          2,1,1   1,1,1   2*LAMBDA;
1,2,1   F1      LAMBDA;            1,1,1   F1      LAMBDA;
3,2,1   3,1,1   2*MU;              3,2,1   F3      SIGMA;
3,1,1   F2      MU;                2,2,1   F3      SIGMA;
2,2,1   2,1,1   2*MU;              1,2,1   F3      SIGMA;
2,1,1   F2      MU;                3,1,1   F3      SIGMA;
1,2,1   1,1,1   2*MU;              2,1,1   F3      SIGMA;
1,1,1   F2      MU;                1,1,1   F3      SIGMA;
```

Table 3. Description of the FEHM for Memory Subsystem

```
PROBABILITIES AND MOMENTS


TRANSIENT RESTORATION EXIT:
  EXIT PROBABILITY: .9800
  FIRST MOMENT OF TIME TO EXIT:  0.
  SECOND MOMENT OF TIME TO EXIT:  0.
  THIRD MOMENT OF TIME TO EXIT:  0.


RECONFIGURATION COVERAGE EXIT:
  EXIT PROBABILITY: .1615e-01
  FIRST MOMENT OF TIME TO EXIT:  0.4500
  SECOND MOMENT OF TIME TO EXIT:  0.2500
  THIRD MOMENT OF TIME TO EXIT:  0.


SINGLE POINT FAILURE EXIT:
  EXIT PROBABILITY: .3850e-02
  FIRST MOMENT OF TIME TO EXIT:  0.
  SECOND MOMENT OF TIME TO EXIT:  0.
  THIRD MOMENT OF TIME TO EXIT:  0.
```

Table 4. Dictionary File Corresponding to 3-Processor, 2-Memory, 1-Bus Example

[The entries are: Component number, name, symbolic failure rate, and FEHM file]

```
1 PROCESSOR       LAMBDA         PROCESSOR.FHM
  INTERFERING COMPONENT TYPES: 1,2
2 MEMORY          MU             MEMORY.FHM
  INTERFERING COMPONENT TYPES: 2,3
3 BUS             SIGMA          NONE
  INTERFERING COMPONENT TYPES: 1,2
```

Table 5. Textual Description of FEHM for LAN Nodes

```
DISTRIBUTIONS AND PROBABILITIES


TRANSIENT RESTORATION EXIT:
 EXIT PROBABILITY:  0.          d+00


RECONFIGURATION COVERAGE EXIT:
 EXIT PROBABILITY:  0.99900000d+00
 DISTRIBUTION TYPE:   EXP
 RATE: 0.16670000d-01


SINGLE POINT FAILURE EXIT:
 EXIT PROBABILITY:  0.10000000d-02
 DISTRIBUTION TYPE:   CONSTANT
 VALUE: 0.    d+00
```

Table 6. Textual Description of FEHM for LAN AWC's

```
DISTRIBUTIONS AND PROBABILITIES

TRANSIENT RESTORATION EXIT:
 EXIT PROBABILITY:  0.          d+00

RECONFIGURATION COVERAGE EXIT:
 EXIT PROBABILITY:  0.99900000d+00
 DISTRIBUTION TYPE:   HYPO
 NUMBER OF STAGES:   3
 RATE: 0.16670000d-01
 RATE: 0.11110000d-01
 RATE: 0.11110000d-01

SINGLE POINT FAILURE EXIT:
 EXIT PROBABILITY:  0.10000000d-02
 DISTRIBUTION TYPE:   CONSTANT
 VALUE: 0.    d+00
```

Table 7. Failure Probabilities and Predicted Availability for 24-Node Network

Time = 144 hr:
Probability [Exhaustion of Hardware] = 1.667×10^{-6}
Probability [Single-Point Failure] = 2.957×10^{-5}
Probability [Near-Coincident Fault] = 1.779×10^{-7}
(ALL INCLUSIVE near-coincident fault rate used)

Instantaneous Availability = 0.9999686
Instantaneous Unavailability = 3.1415×10^{-5}

Table 8. Predicted Unreliability of SIFT System at 10 hr (Summarized Form)
[τ is the time needed to handle a fault]

Cause of Failure	Predicted unreliability for—	
	$\tau = 1$ s	$\tau = 100$ ms
Exhaustion of processors	$0.49860195 \times 10^{-11}$	$0.49860206 \times 10^{-11}$
Near-coincident faults	$0.55500034 \times 10^{-9}$	$0.55500037 \times 10^{-10}$

Table 9. ARCS Failure Data for Short-Term Control Wheel Steering Function

Sensor set: Yaw rate, latitudinal acceleration, normal acceleration, longitudinal acceleration, compass coupler, directional gyro, compass, vertical gyro, control force, air data computer
Failure rate: 762 failures per million hr
Coverage: 0.72

Servo set: Roll, pitch, yaw servos
Failure rate: 390 failures per million hr
Coverage: 0.95

Computer: Multiplex and A/D input, processors and memories, watchdog multiplex and D/A output
Failure rate: 350 failures per million hr
Coverage: 0.95

Hydraulics: 60 failures per million hr
Coverage: 0.95

Table 10. Textual Description of FEHM for Computers in ARCS

[The detection probabilities * ** used are 0.72 for servos and 0.95 for the others]

HARP SINGLE-FAULT MODEL

Time	Distribution and Parameters
ACTIVE transition	Uniform (0,1)
BENIGN transition	Uniform (0,0.5)
Transient lifetime	Exponential (100)
DETECT transition	Uniform (0,0.4)
ERROR transition	Weibull (10.0,2.5)
ERROR-DETECT transition	Weibull (50.0,0.25)
ISOLATION transition	Normal (4.0,1.0)
RECOVERY transition	Erlang (100.0,2.0)
RECONFIGURATION transition	Normal (1.0,0.5)

Other parameters:
Probability of fault detection by self-test: *.**
Probability of error detection: *.**
Probability of isolating detected fault: 1.00
Number of recovery attempts: 5
Probability of successful reconfiguration: 1.00
Fraction of faults which are transient: 0.90
Desired confidence level: 90 percent
Allowable error: 10 percent

Table 11. Failure Probabilities at 10 hr for ARCS Model

[The original design assumed reconfiguration to simplex mode after second failure; the 2/3 design fails on second failure]

	Failure probabilities for—		
Cause of failure	Original design constant coverage	2/3 Perfect coverage	Original design ESPN FEHM
Exhaustion of sensors	8.01×10^{-7}	2.82×10^{-5}	4.38×10^{-7}
Exhaustion of computers	5.14×10^{-7}	1.30×10^{-5}	3.01×10^{-7}
Exhaustion of servos	2.06×10^{-7}	6.62×10^{-6}	1.26×10^{-7}
Exhaustion of hydros	2.05×10^{-10}	1.08×10^{-6}	2.13×10^{-11}
Single-point failure	5.17×10^{-5}	0	5.79×10^{-6}
Near-coincident fault	0	0	1.45×10^{-12}
Unreliability	5.32×10^{-5}	4.90×10^{-4}	6.65×10^{-6}

Table 12. Description of Stages and Basic Events for the Jet Engine Control System

Stage	Basic events	Failure rate
Power supplies	1,2,3	3.00×10^{-5}
Input controllers	4,5,6	1.50×10^{-5}
Data collectors	7,8	7.00×10^{-6}
CPU's	9,10,11	3.26×10^{-5}
1553 buses	12,13,14	1.00×10^{-5}
Output drivers	15,16,17	3.00×10^{-6}
Cross channel data link receivers	18,19,20	4.26×10^{-6}

Table 13. Comparison of Unreliability With Perfect and Imperfect Coverage for the Jet Engine Control System Shown in Figure 16

Time, hr	Perfect coverage unreliability	Imperfect coverage unreliability
1.0	0.27044×10^{-8}	0.10912×10^{-5}
2.0	0.10819×10^{-7}	0.21878×10^{-5}
3.0	0.24349×10^{-7}	0.32898×10^{-5}
4.0	0.43294×10^{-7}	0.43971×10^{-5}
5.0	0.67660×10^{-7}	0.55097×10^{-5}
6.0	0.97448×10^{-7}	0.66277×10^{-5}
7.0	0.13266×10^{-6}	0.77511×10^{-5}
8.0	0.17330×10^{-6}	0.88798×10^{-5}
9.0	0.21937×10^{-6}	0.10013×10^{-4}
10.0	0.27088×10^{-6}	0.11153×10^{-4}

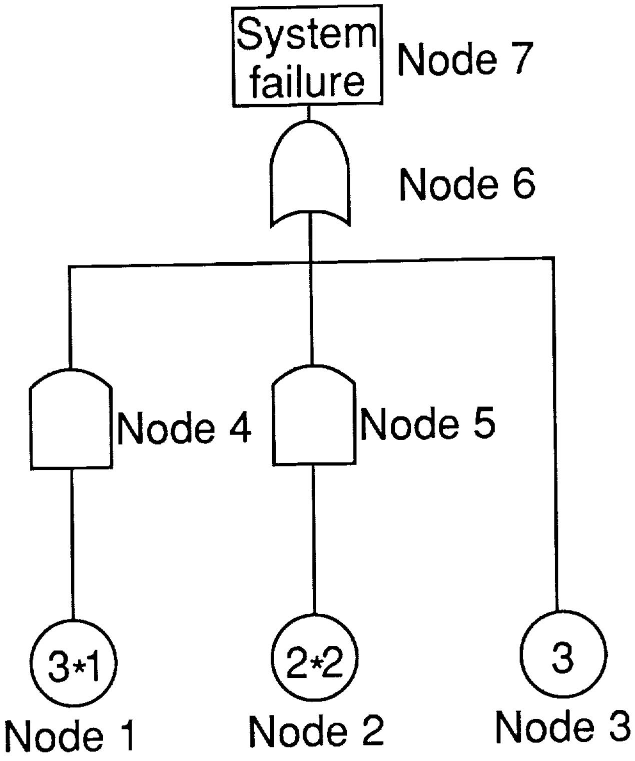


Figure 1. Fault tree representation of a 3-processor, 2-memory, 1-bus system. (A basic event labeled with $i * j$ represents i replications of component type j.)

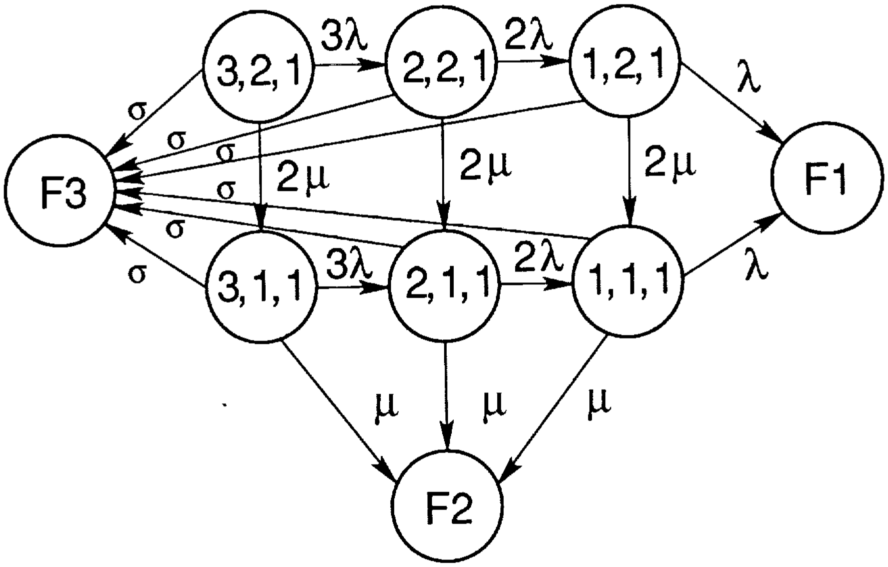


Figure 2. Markov chain representation of the 3-processor, 2-memory, 1-bus system. (λ is the failure rate of the processors, μ is the failure rate of the memories, and σ is the failure rate of the bus.)

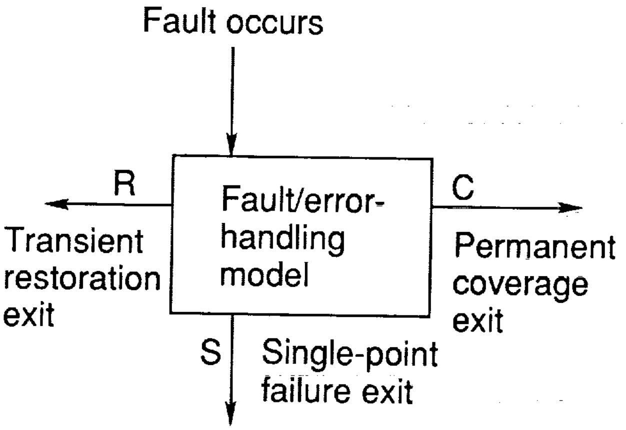
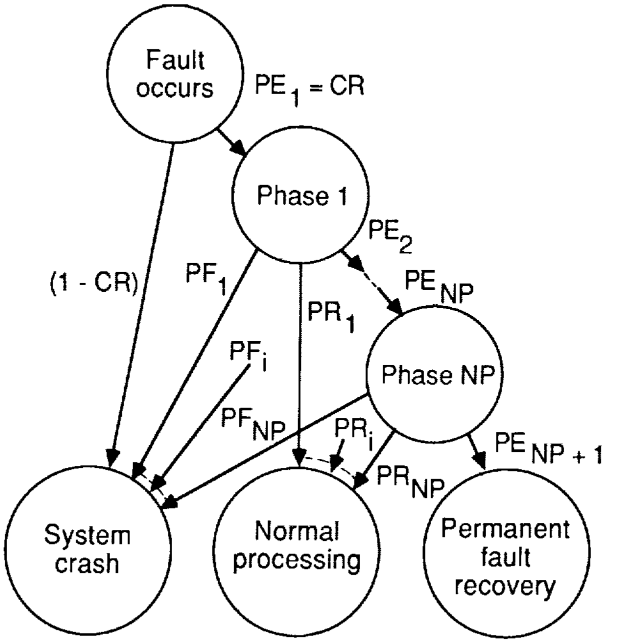

Fault occurs
R
Transient restoration exit
Fault/error-handling model
C
Permanent coverage exit
S
Single-point failure exit

Figure 3. General structure of HARP Fault/Error-Handling Model (FEHM).



Values required by HARP as input -

Probability that fault is transient:	.85
Mean duration of transient fault:	.05
Probability that fault is catastrophic:	.001
Number of transient recovery phases:	6

Phase	Duration	Effectiveness
1	.01	0
2	.02	.1
3	.03	.2
4	.04	.5
5	.05	.7
6	.08	.01

Coverage of permanent fault: .85

Given that a fault occurs, the following parameters are derived from the above HARP input values -

CR probability that fault is noncatastrophic
PF probability that system crashes
PE probability that system enters recovery state
PR probability that transient recovery is successful

Figure 4. The ARIES transient-fault recovery model and parameterization for processors.

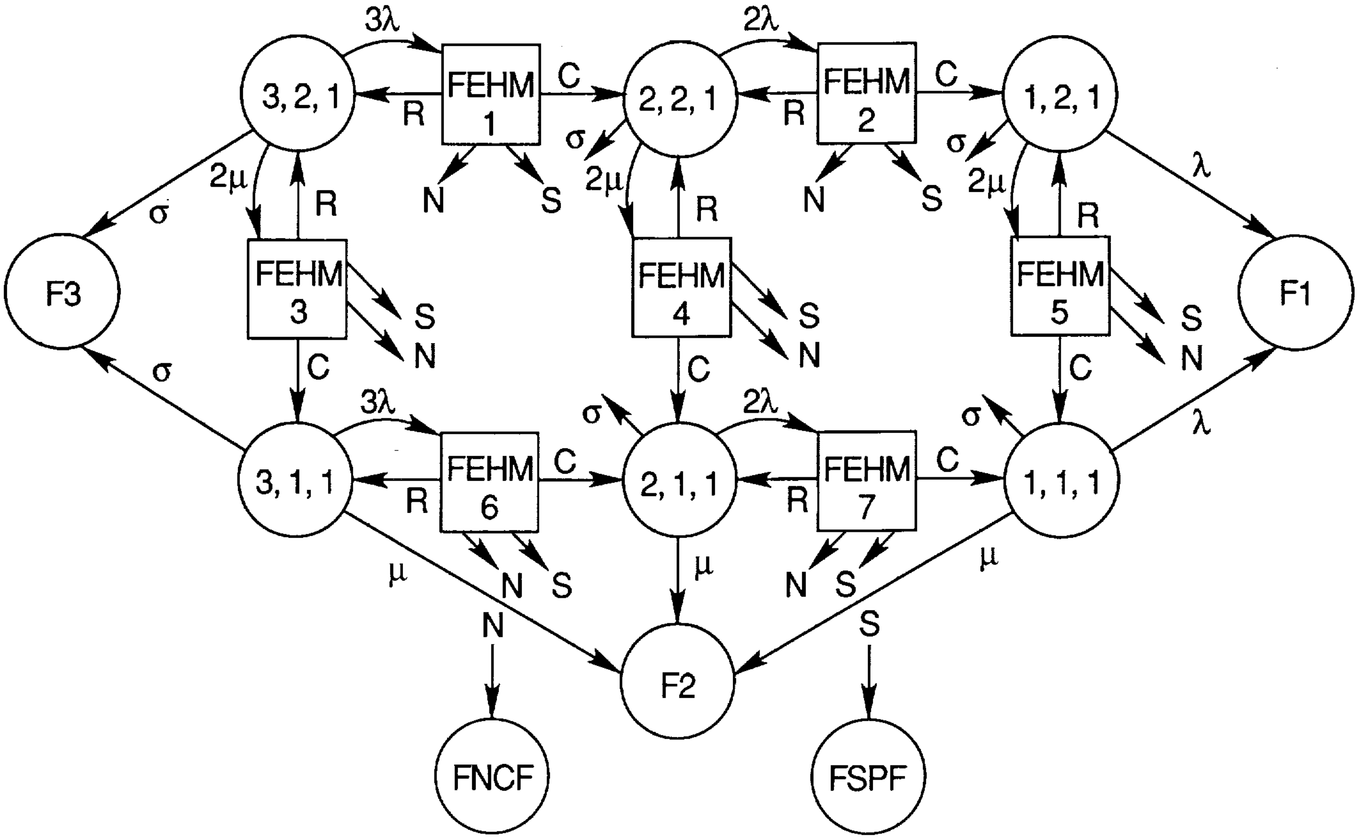


Figure 5. Automatic insertion of a FEHM for failures of redundant components.

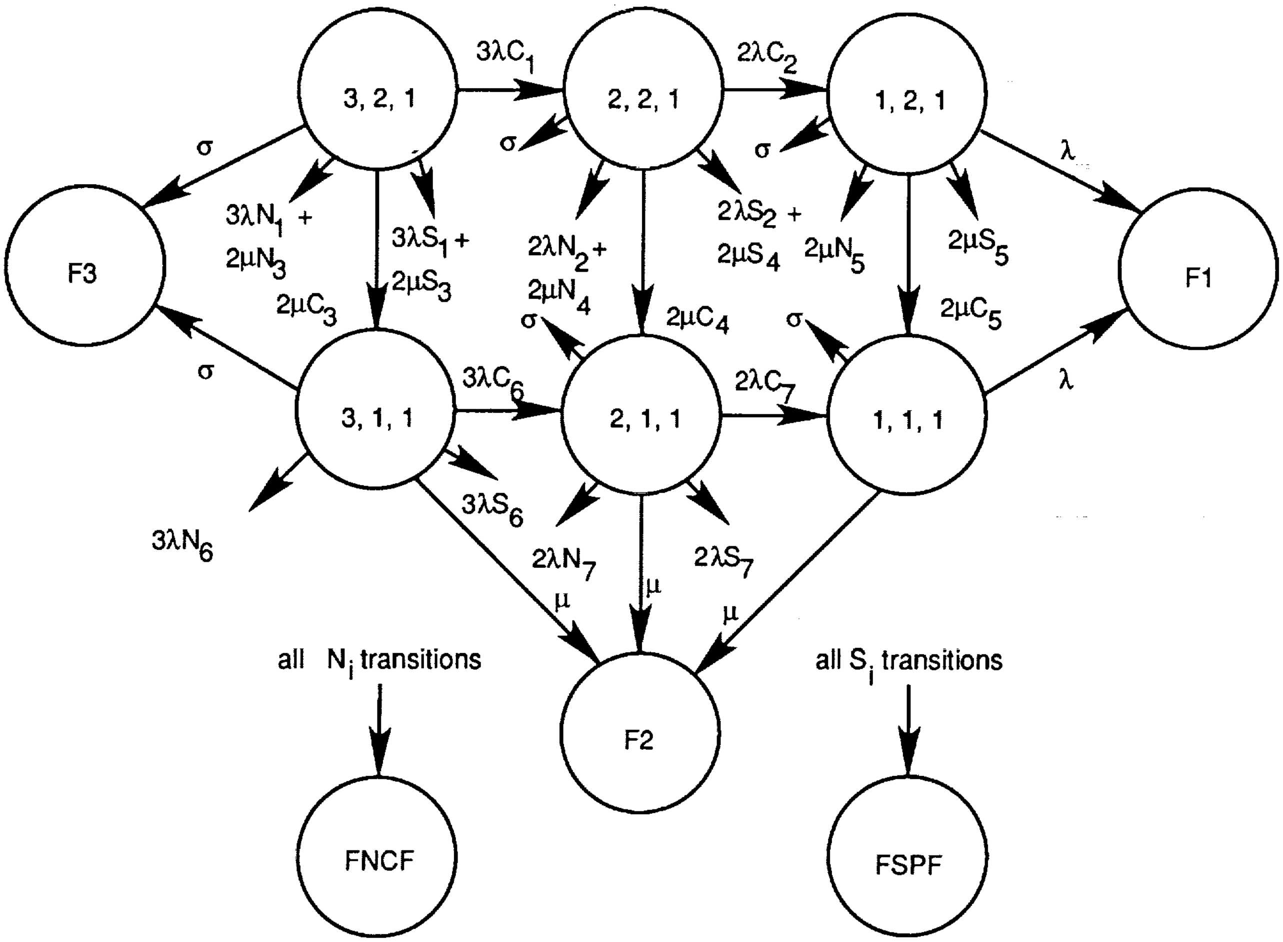


Figure 6. "Imperfect coverage" Markov chain corresponding to figure 5.

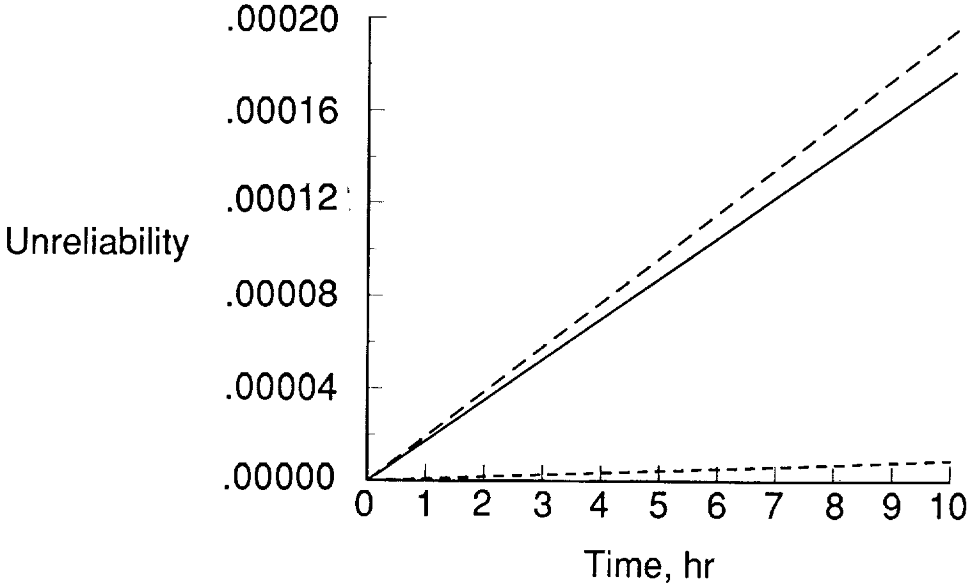


Figure 7. Predicted unreliability for 10-hr mission of 3-processor, 2-memory, 1-bus system.

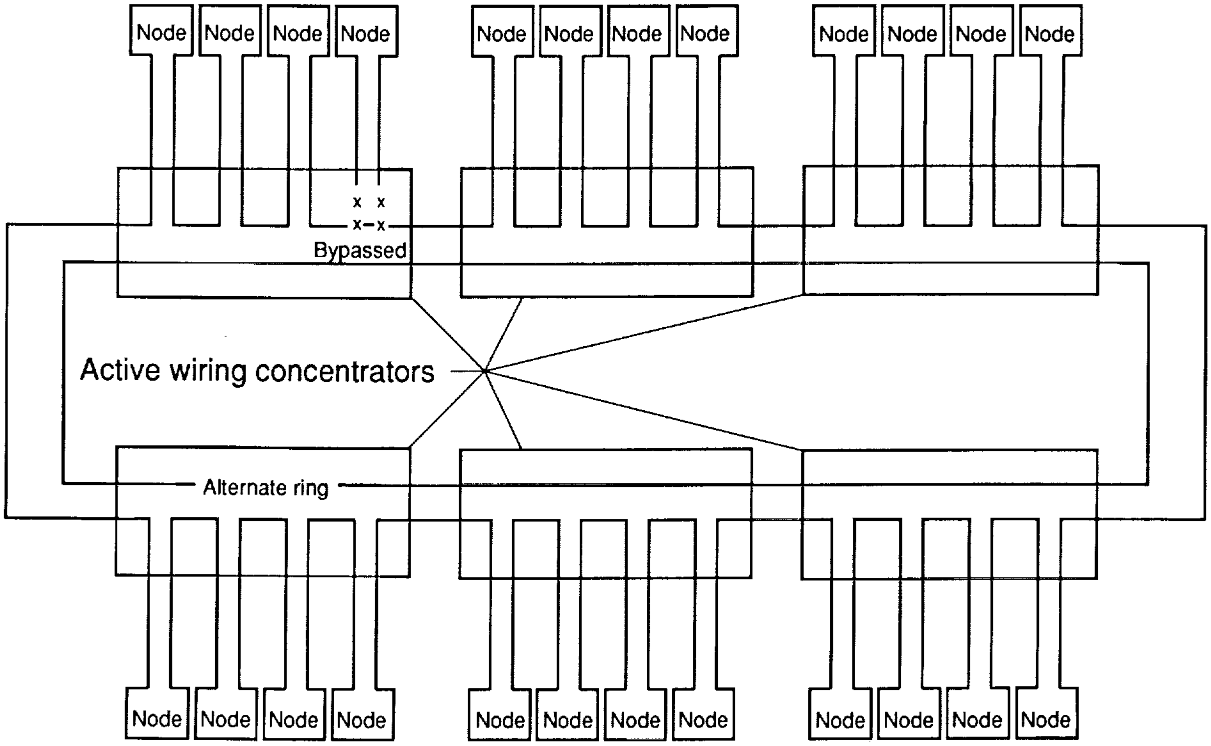


Figure 8. A token-ring architecture for token-ring networks.

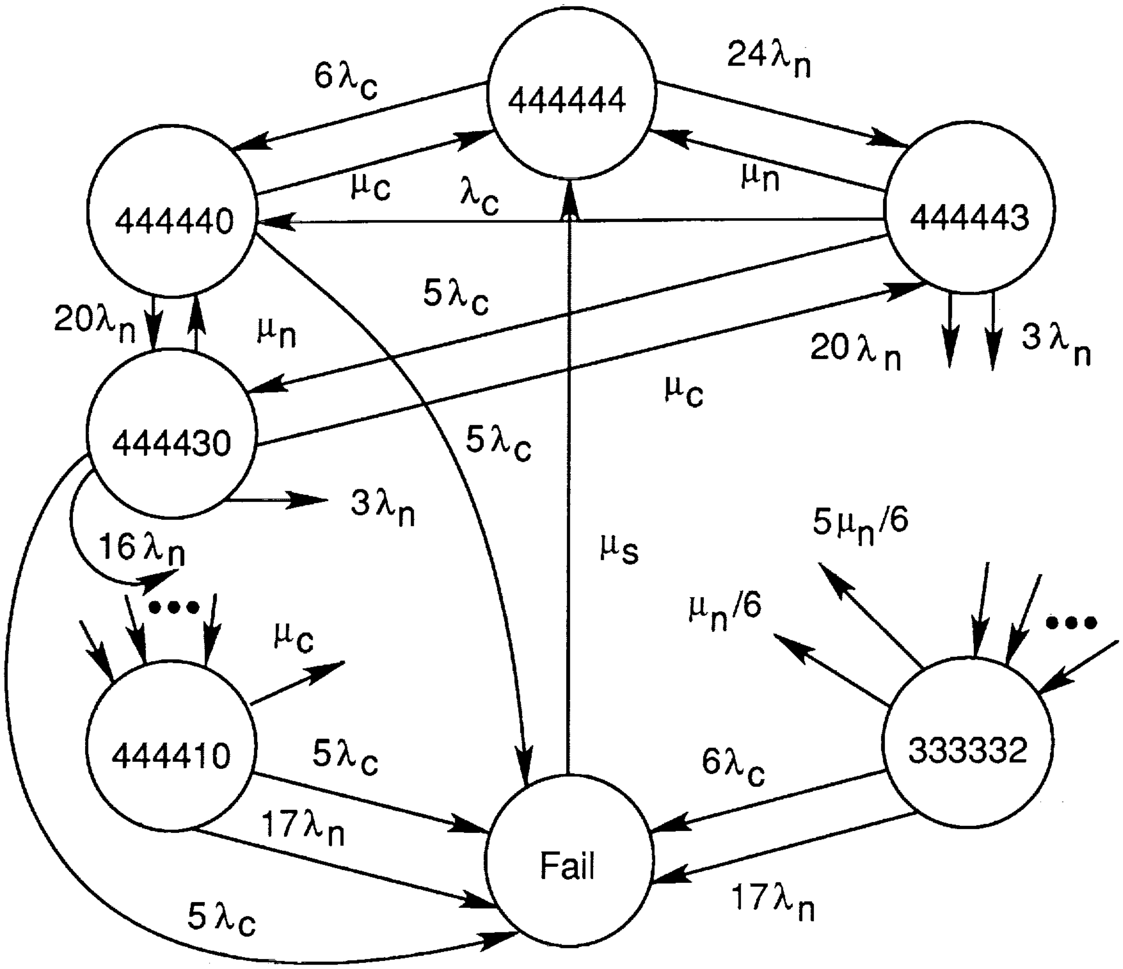


Figure 9. Markov chain representation of token-ring network.

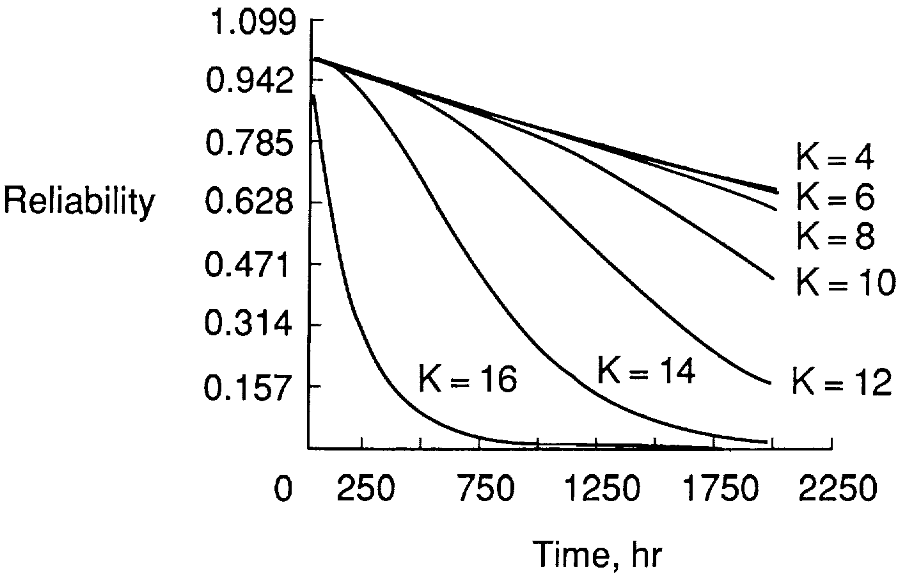


Figure 10. Reliability of C.mmp system with perfect coverage for varying K. (K is the number of processors and memories.)

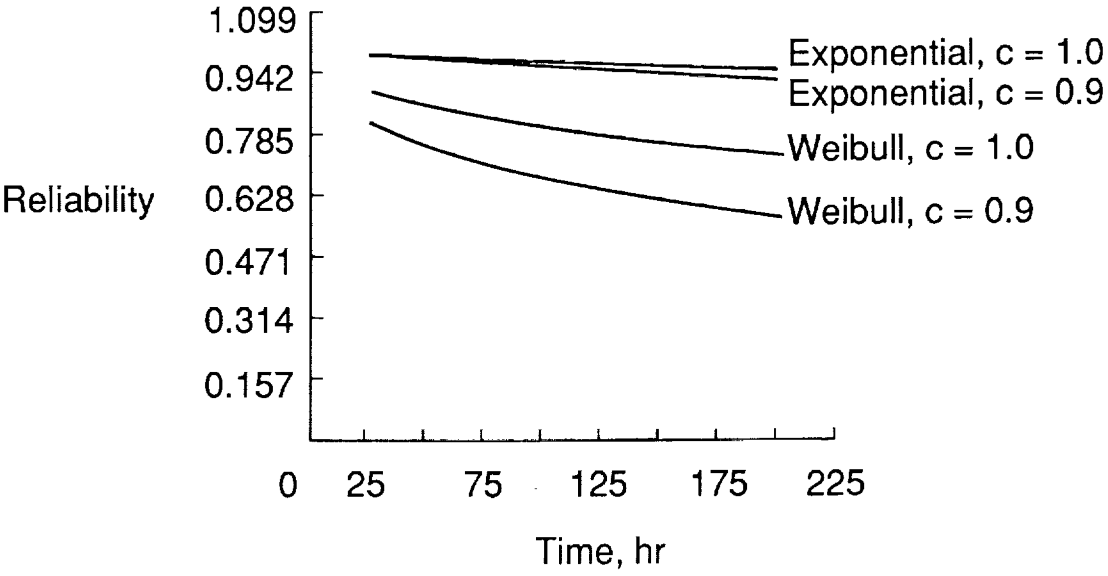
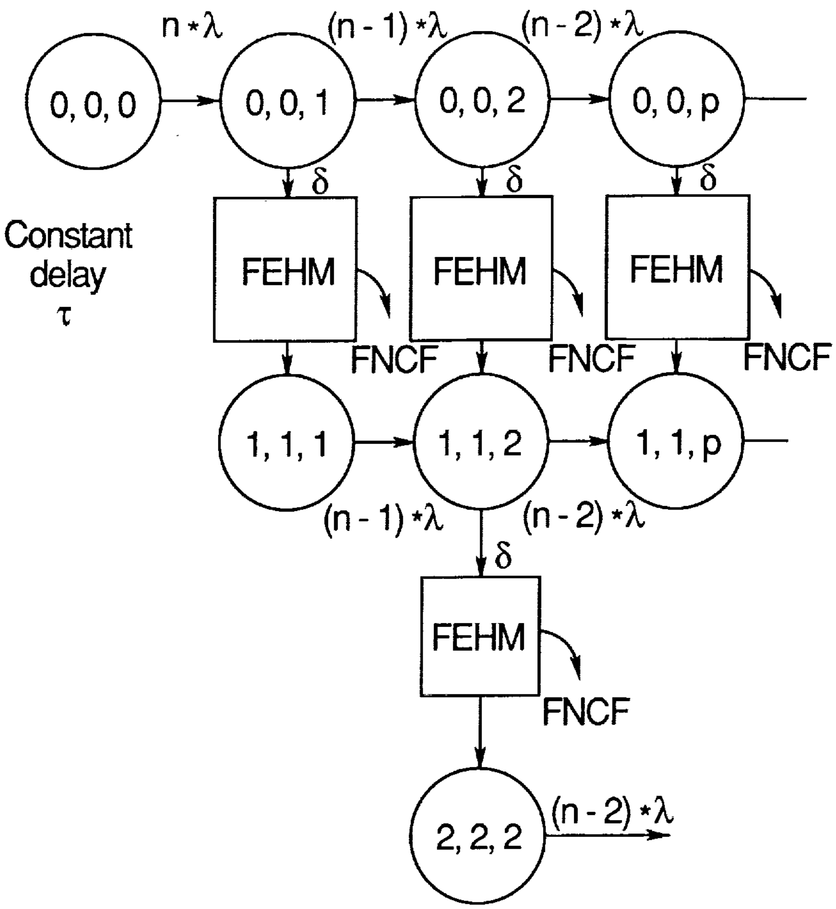
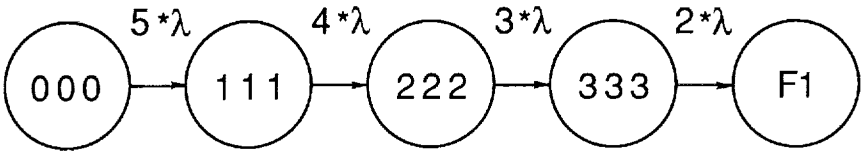


Figure 11. Comparison of C.mmp system ($K = 4$) with exponential and Weibull failure distributions with perfect coverage ($c = 1.0$) and without perfect coverage ($c = 0.9$).



(a) Permanent fault SIFT model. (δ is the detection rate; τ is the time needed to handle a fault.)



(b) FORM input to HARP after removal of instantaneous states.

Figure 12. The SIFT system.

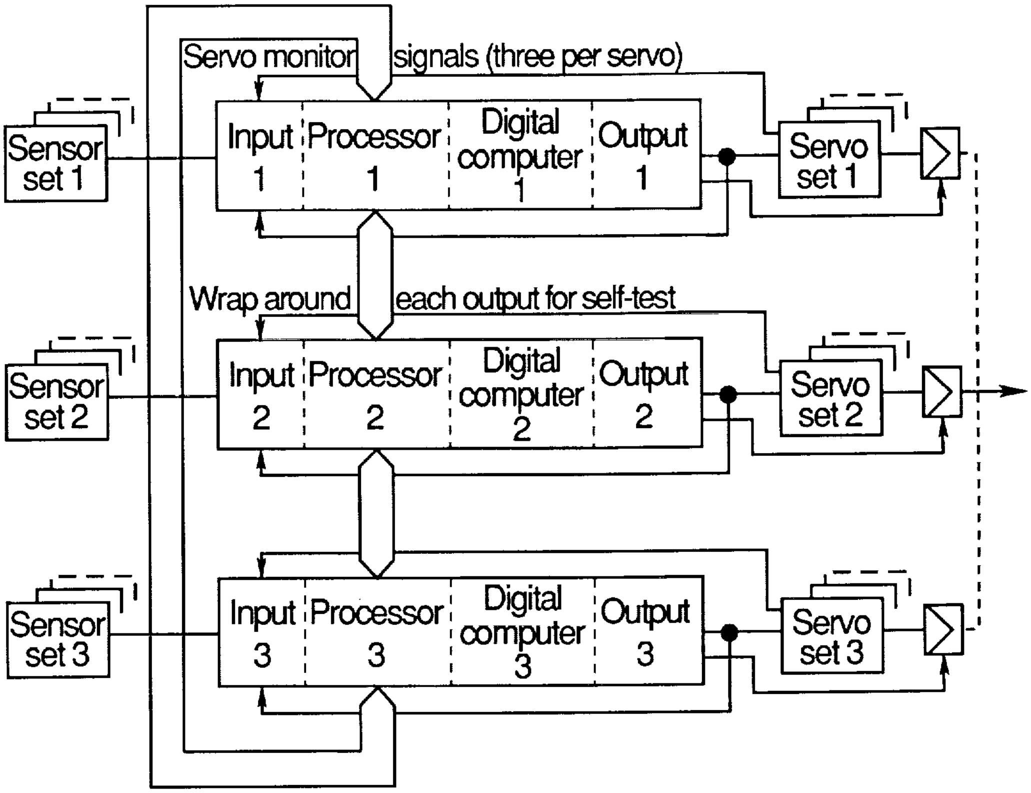


Figure 13. Advanced reconfigurable computer system (ARCS).

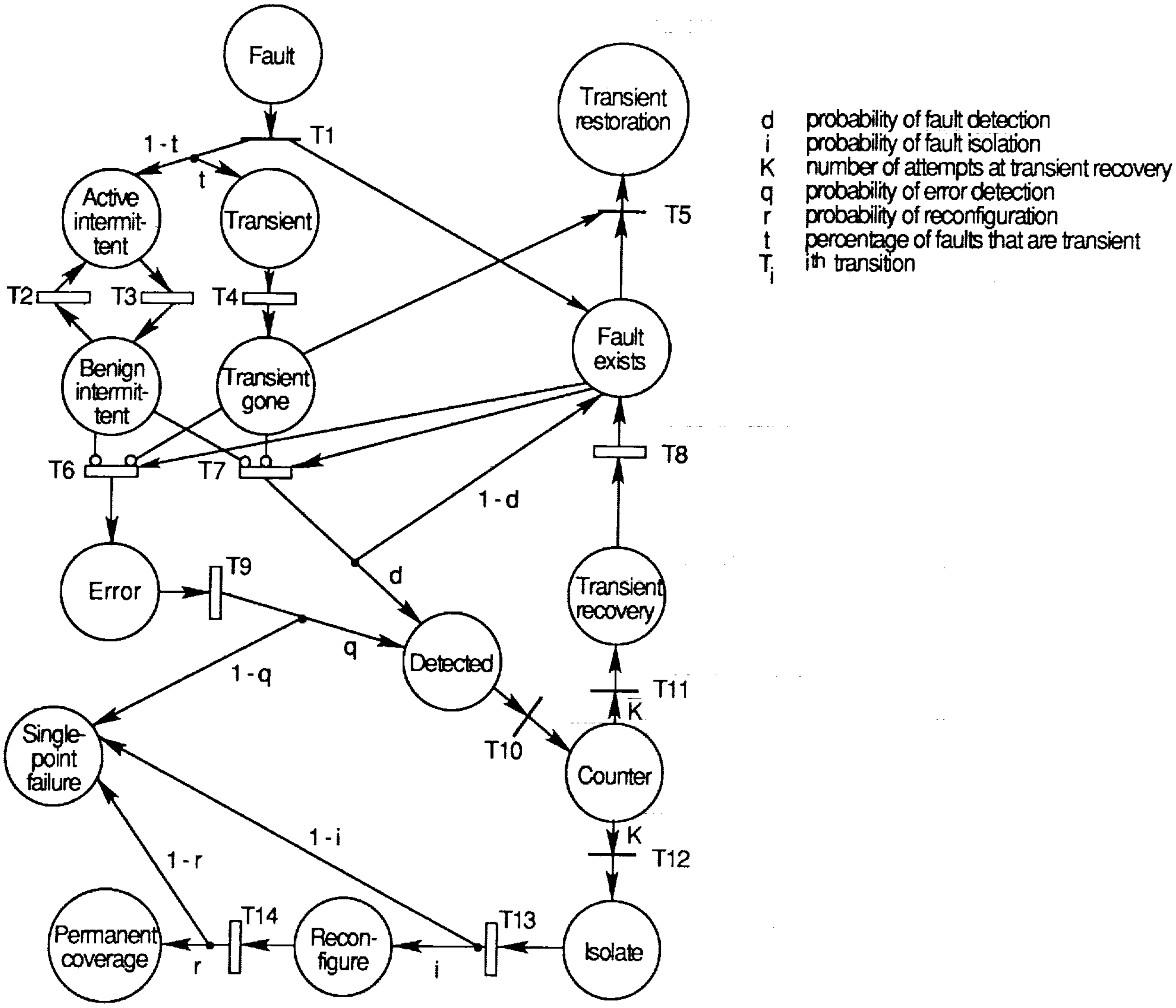


Figure 14. The HARP ESPN single-fault model. T1, T5, T10, T11, and T12 are instantaneous transitions; all other transitions may be assigned a holding time distribution.

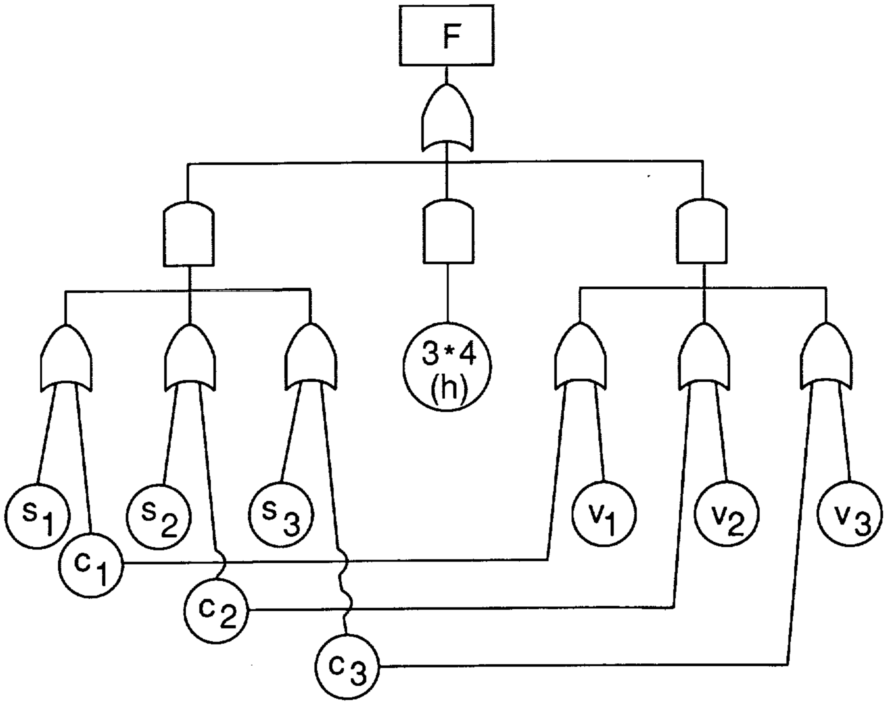


Figure 15. Fault tree representation of ARCS system.

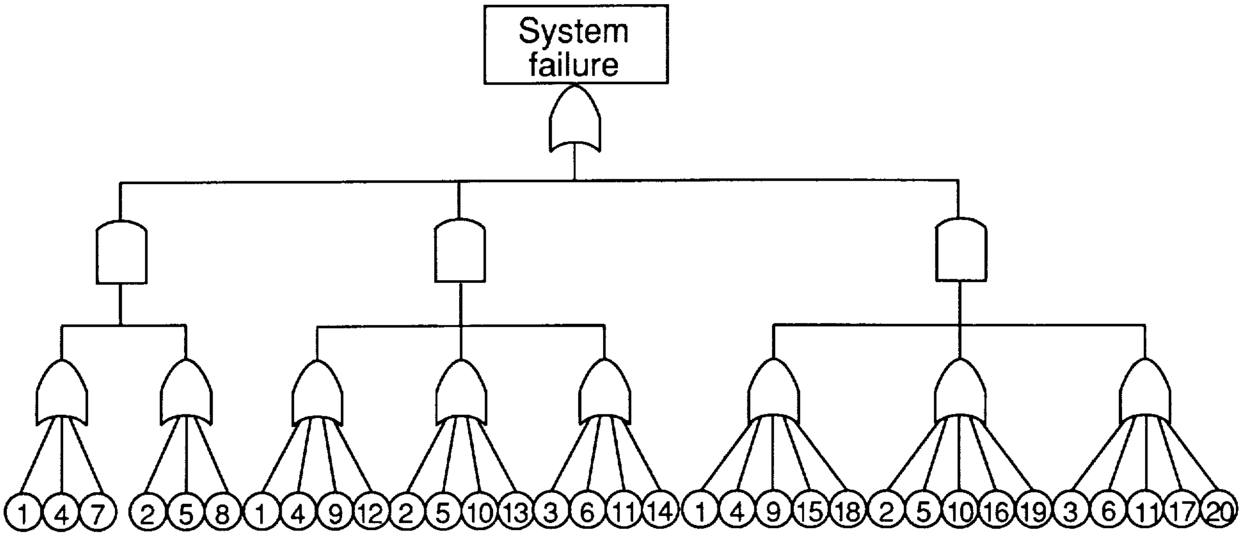
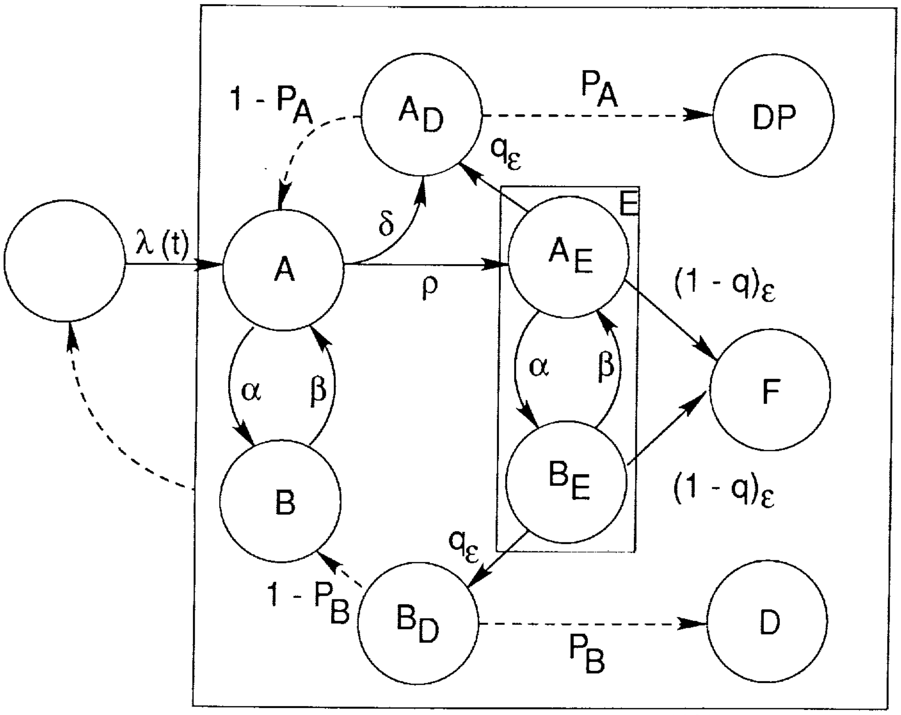


Figure 16. Fault tree representation of a fault tolerant jet engine control system. (If two basic event labels are the same, they represent the same component.)



Parameters, assuming all faults are permanent
(those not listed are 0.0)

Fault Detection Rate δ = 360 per hr
Error Propagation Rate ε = 3600 per hr
Error Production Rate ρ = 180 per hr
Reconfiguration Probability P_A = 1.0
Error Detectability q = 0.97 or 0.99

$1/\alpha$ transient or intermittent active duration time, given transition only to state B or B_E

$1/\beta$ intermittent benign duration time

Figure 17. Markov version of the CARE III single-fault model. (The error detectability is 0.97 for data collectors and 0.99 for the other stages.)

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No. NASA TP-2760	2. Government Accession No.	3. Recipient's Catalog No.

4. Title and Subtitle Applications of the Hybrid Automated Reliability Predictor—*Revised Edition*	5. Report Date December 1988
	6. Performing Organization Code
7. Author(s) Salvatore J. Bavuso, Joanne Bechta Dugan, Kishor Trivedi, Beth Rothmann, and Mark Boyd	8. Performing Organization Report No. L-16304
9. Performing Organization Name and Address NASA Langley Research Center Hampton, VA 23665-5225	10. Work Unit No. 505-66-21-02
	11. Contract or Grant No.
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546-0001	13. Type of Report and Period Covered Technical Paper
	14. Sponsoring Agency Code

15. Supplementary Notes

Salvatore J. Bavuso: Langley Research Center, Hampton, Virginia.

Joanne Bechta Dugan, Kishor Trivedi, Beth Rothmann, and Mark Boyd: Duke University, Durham, North Carolina.

An earlier version of this paper appeared in IEEE Transactions on Reliability, June 1987.

16. Abstract

The Hybrid Automated Reliability Predictor (HARP) is a software package that implements advanced reliability modeling techniques. In this paper we present an overview of some of the problems that arise in modeling highly reliable fault tolerant systems, loosely divided into model construction and model solution problems. We then describe the HARP approach to these difficulties, which is facilitated by a technique called behavioral decomposition. The bulk of this paper presents examples of the evaluation of some typical fault tolerant systems, including a local area network, two fault tolerant computer systems (Carnegie-Mellon University multiprocessor system C.mmp and Software Implemented Fault Tolerance (SIFT)), and two examples of flight control systems.

17. Key Words (Suggested by Authors(s)) HARP Reliability Coverage Fault tolerance Fault handling	18. Distribution Statement Unclassified–Unlimited Subject Category 61

19. Security Classif.(of this report) Unclassified	20. Security Classif.(of this page) Unclassified	21. No. of Pages 27	22. Price A03

For sale by the National Technical Information Service, Springfield, Virginia 22161-2171